CONCEPT OF OPERATIONS

1. Introduction

1.1. Scope

This document contains the concept of operations for the Checkout and Launch Control System (CLCS) and will provide system level requirements as identified by the user community for the Operations Control Rooms (OCR), specialized processing sites, and all activities required to define, prepare and run test operations.

1.2. Operational Overview

Shuttle processing at KSC can be categorized in two ways: Day to day flight element and ground support equipment processing and integrated vehicle processing which includes launch. These categories of processing have unique requirements which affect how we use CLCS.

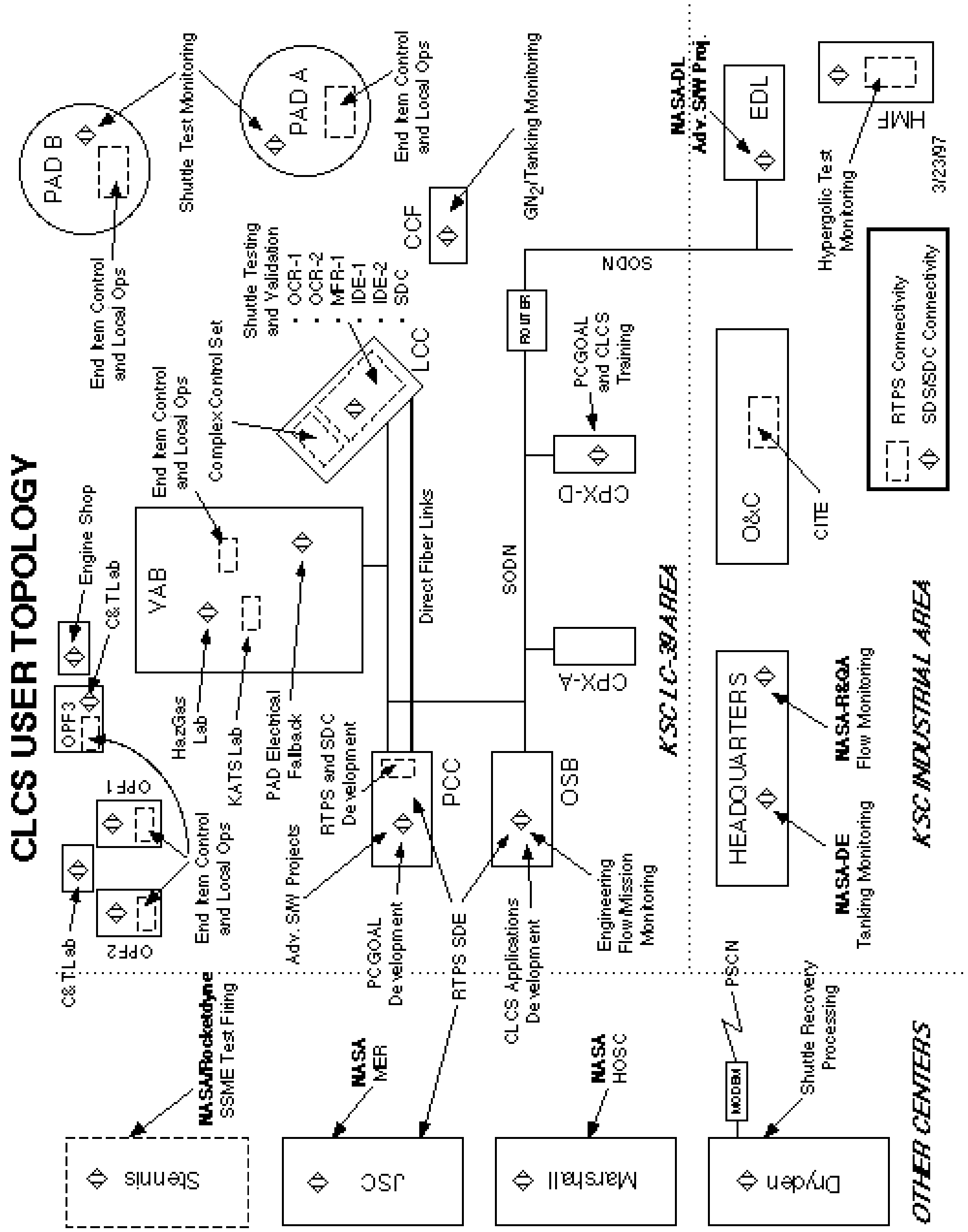
Day to day processing involves use of multiple control rooms, CITE, HMF (Hypergolic Maintenance Facility), and CCS resources. On any given day, a control room would typically be divided into two flow zones processing a non integrated vehicle, the associated GSE, and maybe an MLP on one side while the other side may simply have a need to process an ET housed in VAB high bay 2 checkout cell, and the GSE at the non-launch pad. This may be occurring in two control rooms on the same day with a mixture of flight elements, GSE and facilities. Engineers are controlling their tests from the console workstations in the control room or on-site locally with a plug in portable workstation. They may also be in their office retrieving SDS data for analysis from their office.

Off-line processing of different elements occurs daily at the HMF for the OMS pods and FRCS and CITE for payload checkout. The Complex Control Center monitors the facility functions like HVAC, electrical power, Firex, and pneumatics around the clock.

An orbiter becomes integrated in the VAB when it is mated to its external tank and boosters. At this point this integrated vehicle, the associated GSE and the launch pad systems would be controlled out of a single control room. This control room configuration would remain through launch. Typically the OPF bay from where the orbiter came remains in the TCID though launch.

During the checkout and launch of the integrated vehicle on the launch pad, data is piped to various places for review and analysis. This includes the HOSC at Marshall, the mission evaluation room at JSC, etc. Once the vehicle is successfully on orbit, telemetry data is downlinked to JSC and distributed to other field centers including KSC.

Figure 1-1 illustrates the wide distribution of CLCS data required across the center and the agency.



1.3. Document Overview

This document is identified as the CLCS Concept of Operations. The document is organized into six sections. Section One contains the introduction of the operations concept and an operational overview of CLCS. Section Two contains definitions of CLCS physical characteristics and roles and responsibilities. Section Three is separated into two subsections, Support Operations and Test Operations, which describe how CLCS will be used for all aspects of operations. The fourth section discusses system processes, including hardware, system hardware, and application software. The User Environment section describes the human computer interface (HCI) capabilities necessary support test operations in the OCR's. The final section contains operations scenarios to cover situations from local operations up to, and including, launch and landing.

1.4. Applicable Documents

List necessary documents

2. Operations Concepts

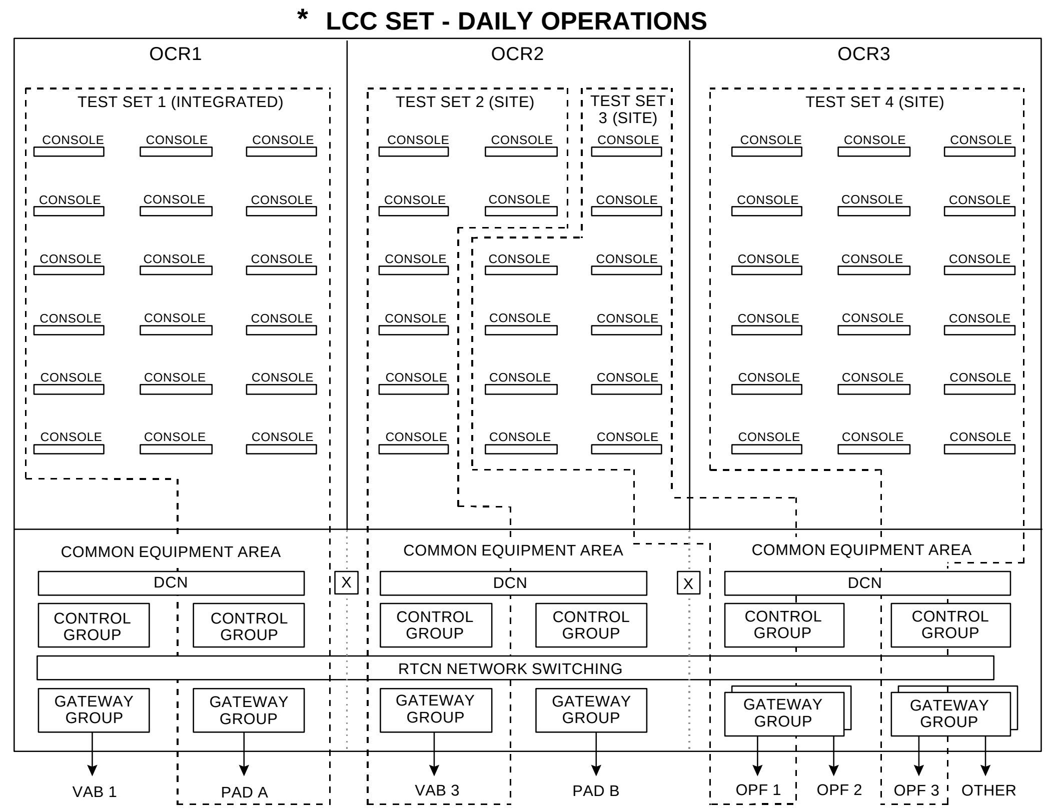
This section will discuss the physical characteristics of the CLCS areas which includes the LCC Set, specialized processing sites, local operations, and the software sustaining environment. The section will also discuss the roles and responsibilities of personnel supporting CLCS.

2.1. Physical Characteristics

2.1.1. Launch Control Center (LCC) Set

The new OCR is a CLCS area capable of providing command, control and monitoring of ground and vehicle processing for an entire mission, from landing to launch. CLCS will have the flexibility to meet the changing demands of all aspects of shuttle processing in a quick and reliable manner. The room is configurable into flow zones to support multiple flows, if necessary. Typically, a control room will only be divided to support 2 non-integrated flows although the rooms can be divided to meet varied processing requirements. Figure 2-1 illustrates one option for normal day to day operations. From the start of integrated operations, the Shuttle Interface Test through Terminal Countdown and Launch or Abort/Safing and Scrub/Turnaround, an entire control room will normally support the prime vertical flow. For launch countdown and certain launch countdown simulations both prime OCR's will be configured to support the prime vertical flow. Figure 2-2 illustrates the planned launch countdown configuration.

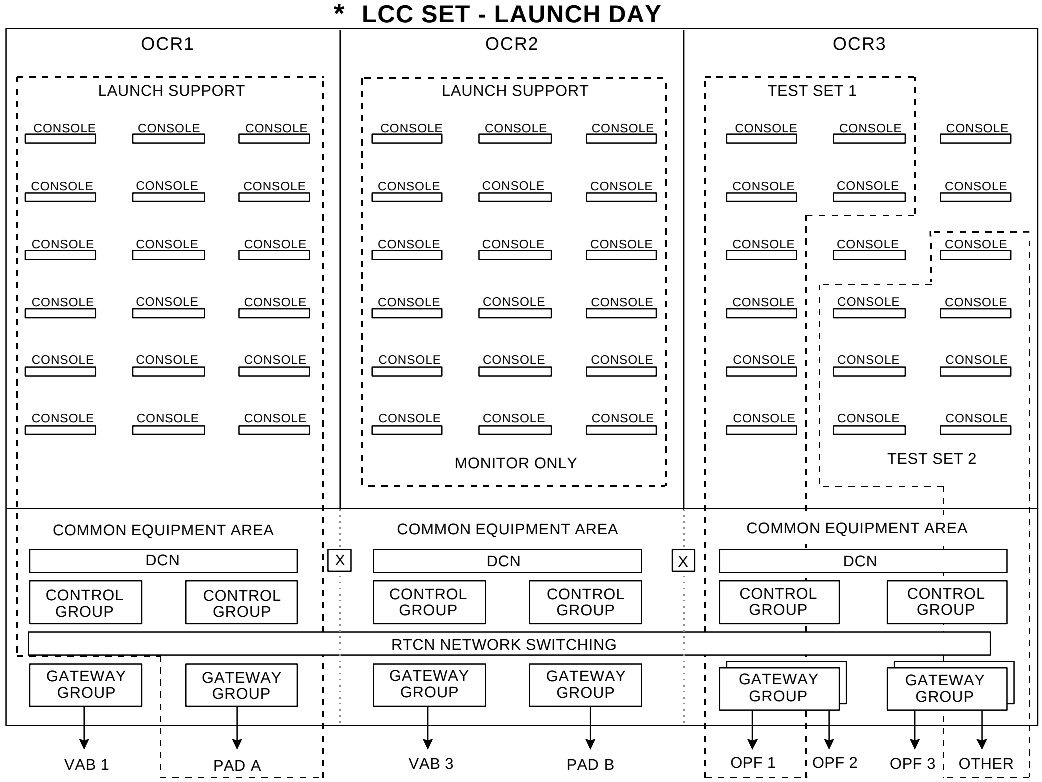
The LCC Set will be made up of systems engineering type consoles, TC type consoles, and the common equipment area. Each test set within the LCC Set will provide sufficient hardwire links to safe the vehicle and its environment (Pad, OPF, etc.) completely independent of CLCS. The area will also provide dedicated printer support within headset cord range of all positions. Sufficient storage assets and table surface area will be provided for auxiliary data, books and equipment.



* Not representative of actual number or layout of consoles in OCR

Figure 2-1 Daily Operations Configuration

* LCC SET - LAUNCH DAY



* Not representative of actual number or layout of consoles in OCR

Figure 2-2 Launch Day Configuration

2.1.1.1. Flow Zone

The CLCS Flow Zone concept provides the capability to command, control and monitor multiple processing flows or TCIDs within the same control room. The control room is divisible into distinct areas for each TCID/processing flow. Command paths are logically isolated to those areas which support that flow. The area will contain physical clues (i.e., signs, display headers, etc.) as to which flow an operator is working on and can be configured on an as needed basis to provide sufficient consoles for testing associated with that flow. Sufficient storage assets and table surface area will be provided for auxiliary data, books and listings. Storage assets may be movable to facilitate area reconfiguration. The area will provide sufficient output assets (printers, faxes, copiers, trip charts) to prevent interference between the systems testing within a flow and between Flow Zones.

2.1.1.2. System Engineer Consoles

With the CLCS concept, the definition of “Console” changes somewhat. Consoles are no longer associated with system discipline but, because all displays are resident at every location, consoles become generic pieces of hardware that can support any engineering system’s testing on any given day.   Systems will still be consistently grouped within a control room on a day to day basis for normal test operations whenever possible but there will be a standardized configuration identified for major test operations. In addition a console is more of an engineering workstation that can serve as one or two distinct positions. It contains updated technology to display data and incorporates separate command, monitoring, and business information networks. There will be a dedicated area on the console for OIS, safing panels, and other institutional legacy type equipment.

Engineering disciplines will have unique command application sets. (similar to today's 27 application sets) Any command console position should be capable of running at least 4 of these command applications sets simultaneously without significant loss of performance. Engineers should also be able to command a single application from multiple positions. The Console CPU's perform the function of the Human Computer Interface (e.g. Display generation, Operator Input and system interrogation functions). In this context, the HCI initiates and attaches to application elements located elsewhere (e.g. common equipment area), so that the application set specific software unique to a command/monitor console position is minimized.

In addition, all systems within a flow zone will have the capability to view any command display without risk of sending an inadvertent command. This includes viewing command displays for another system from the initial system's command monitor. There is a requirement for the ability to restrict commanding to a specific system or group of systems. In addition, a minimum 2 step process to execute a command in the application software is required. Software standards should be written to preclude automatically issuing commands or setting limits when an application is called up, rather these should be initiated by some operator action.

If we continue to do business as it is done today, management and support personnel will reside in an operational environment for launch (i.e., FR-2 is used today). These personnel should be able to view the same displays on their command screens that are being viewed on the launch room command screens, but be restricted from initiating commands.

2.1.1.3. Test Director/Conductor Consoles

Test Directors and Test Conductors serve a purpose in the firing room unique from that of systems engineers. Provisions must be made in these consoles for such things as expanded OIS and OTV capability, RF OIS capability, Astronaut Communication panels, Paging & Area Warning Systems, GLS hold switch, Emergency Camera Panel, Bird Deterrent Panel, and multiple phones. There are no differences that represent system architectural drivers since these consoles require access to the same networks used by systems engineers.

There are, however, different configuration requirements for the TC consoles based on specific job functions. Some of the TC consoles need the capability to enable or disable local drops and to alter control room configuration during day to day operations (i.e., increasing or decreasing the number of workstations in a flow zone). In addition, certain TC consoles should have the same command and control capability of a system console.

2.1.1.4. OMR/OSR Consoles

Program and Center management support of launch countdown from the OMR and OSR will require a different console configuration than the rest of the firing room. This area should have access to the real-time data and informational data networks along with OTV and OIS. Each position will have a dedicated phone.

Only one OMR/OSR area will be built requiring a total of 36 positions. This area will be accessible from both launch control rooms.

2.1.2. Local Operation Definition

Hands on, face to face communication with on-site personnel is an important improvement to the control of operations. The ability of an engineer on the floor to directly monitor operations and command troubleshooting on-site is a useful tool. There is no plan to establish satellite control rooms, the purpose of local operations is to eliminate the need to have two engineers support a task. It will also allow closer coordination with the technicians working the job.

CLCS local operations will include the ability to monitor and command flight and ground hardware from multiple places in all processing sites. This will be implemented by extending the required networks to all processing sites. Each site will have network drops installed near vehicle processing locations. The number of drops will initially be limited to key locations such as the crew module, near the aft access door, at the LOX and LH2 storage facilities, the MLP's and in the PTCR. Additional drops will be installed to meet specific engineering requirements at a later date.

Until specifically required by an engineer on site, all network drops at a local site will be enabled for monitor only. If command capability is needed (limited to the activity authorized for that location), activation by the Test Conductor will be required. Positive identification of user location will be required.

Portable workstations will be required that can provide monitoring, command, limited advisory systems access, and limit checking capability. Although these work stations will only need a limited set of the capabilities of an OCR workstation, they must support both single commands and command sequences. They will not be required to support OTV interfaces or access to the institutional network. We accept that there will be some delay in command transmission/response times, therefore, time critical commanding does not need to be supported.

Use of local control for hazardous operations will be limited. Local control of operations will not be allowed if loss of command capability increases the hazard, or if potential exists for a hazardous area to expand to include the control location. Restrictions also apply for operations requiring multiple system integration/coordination.

2.1.3. Specialized Processing Sites

Specialized Processing Sites are defined at those areas outside the OCR's where CLCS activities will be performed. These include the HMF, CITE, SAIL, DFRF, and CCS. These sites will require both Systems Engineer and TC type consoles to support testing requirements. CLCS will have the same capabilities at these sites as identified for the OCR's.

The specialized processing sites will have either one or two test sets installed depending on the test requirements for that site. The generic test set should be used as a baseline to identify equipment required for these sites.

2.1.4. Software Sustaining Environment

SDE (Software Development Environment) and IDE (Integrated Development Environment) are areas outside the OCR's where CLCS hardware and software are developed and tested prior to release for use in actual test operations. There will be at least four SDE and at least two IDE areas necessary to support hardware and software development.

The SDE will include fixed locations and a desktop development environment. These areas will provide all capabilities necessary to emulate the CLCS system to facilitate testing efforts. The four SDE areas are:

1. SDE-1: This set will be used for the development of system software and hardware in a test environment with informal configuration control. It will consist of development and operational workstations, networks, Data Distribution Processors (DDP), Command Control Processors (CCP), and gateways.
2. SDE-2: This set will be used for the development of system software and hardware in a test environment under formal configuration control. It will also consist of development and operational workstations, networks, DDP's, CCP's, and gateways.

3. SDE-H: This set will be used for the development of system software in Houston in a test environment with informal configuration control. It will also consist of development and operational workstations, networks, DDP's, CCP's, and gateways.
4. SDE-Users: This will be a set of development workstations in the PCC and OSB to permit the development of software and associated documents in an office environment. This environment is supported by Configuration Management servers located in SDE-1, SDE-2, and the SDC. Although the workstations will be located in the PCC and OSB, the servers will be in a controlled environment.

In addition to the SDE areas, there will be a Simulation Development Environment that will provide a desktop capability supported by SDE-Users. The simulation development environment will provide a stable execution system and configuration to develop and sustain math model applications.

The IDE will provide a near-operational system where software can be tested against a stable hardware and system software configuration. In order to obtain the maximum benefit of the IDE, if must have all development tools and languages available during validation and system checkout. The two IDE areas are:

1. IDE-1: This area will be a set of actual CLCS hardware in a test environment under formal configuration control. The set will configured just like hardware deployed in an operational environment to support application validation.
2. IDE-2: This area will be a redundant set of actual CLCS hardware in a test environment under formal configuration control configured as a real set to allow total system checkout.

2.2. Roles and Responsibilities

2.2.1. Users

Users are divided into three groups defined by the types of tasks they perform. The first task is test preparation. Users who support test preparation include Test Application Developers, Test Data Base Developers, Test Configuration Builders, and Test Configuration Managers. Preparation consists of those tasks which are required to prepare all aspect of test definition and configuration.

The second task is test execution. Users who perform test execution tasks include System Test Engineers and Test Conductors. Execution consists of control and supervision of CLCS test operations.

Users who perform Support Administration tasks include System Administrators, System Configuration Managers, and System Data Base Administrators. Administration consists of account maintenance, configuration management, and maintenance of system data bases.

2.2.2. Operators

Operators are divided into two groups defined by the types of tasks they perform. The first task is Support Operations. Personnel who perform Support Operations tasks include Set Managers, Flow Zone Managers, System Engineers, Support Operators, and Network Managers. Support Operations consist of management and control of CLCS system resources and preparation and monitoring of each CLCS TCID in support of testing.

The second task is Support Maintenance. Personnel who perform Support Maintenance tasks include Maintenance Engineers and Maintenance Operators .Maintenance consists of failure analysis, fault isolation, recovery, and preventative maintenance.

2.2.3. Sustainers (Developers)

Sustainers are the personnel which provide support to keep the Hardware, System Software, Applications Software and Data Bank/System Build configured properly and to update those configurations as necessary.

3. CLCS Operations

This section will discuss both support and test operations. Support operations are those activities that provide the framework for test operations. Test operations are those activities which are directly responsible for controlling and monitoring tests with CLCS.

3.1. Support Operations

Support Operations include all of the activities and tasks to define and prepare for the execution of Test Operations and to provide auxiliary services, as required.

3.1.1. Test Definition

<u>CLCS Data Bank.</u> The Shuttle Data files form JSC and GSE Data files from KSC are used by the DBSAFE process to update the CLCS Data Bank prior to beginning the test definition phase. The Data Bank contains detailed descriptions of all Function Designators or Measurements to be monitored or controlled by the CLCS. The Data Bank resides in the SDC.

<u>TCID Build</u>**.** A TCID is a compilation of all software, tables, files, and parameters needed to configure a CLCS Test Set to support a specific Test Operation. These tables and parameter files are also used to configure required SDC and SIM processes. The TCID Build process, performed on the SDC,

3.1.2. Resource Allocation

The Resource Manager and/or Activity Manager, operating in the SDC, are used to identify, select, and allocate appropriate hardware and software resources.

3.1.2.1. Activity Management Task

The loading of software to resources within a set is managed by the Activity Management task. When invoked by the Master Function, this task provides the capability to load an entire CLCS set, a specified group of resources within a set, or an individual resource.

When an entire set is to be loaded, a test configuration (which consist of an associated collection of software and hardware resources needed to support a test), is pre-defined and stored in a centralized storage location within the SDC. The software to be loaded consists of the operating system, system applications and user engineering applications, as well as supporting tables, etc., that make up the test configuration software package (TCID). The software test package is built by the software development process and placed under configuration control prior to being loaded into an operational environment. When a configuration is needed for testing, operations personnel retrieve the specific configuration and, upon verification of the desired configuration, initiates the autonomous loading of the specified hardware. The Activity Management task associates a group of gateways, processors, servers, etc., with a software test package as pre-defined by the Test Configuration, in addition to a group of user positions (workstations) to support the test configuration. The Activity Management task monitors the progress of the load process, while the SDC loads the hardware and provides data integrity and verification mechanisms (CRC, checksums, etc.) to insure proper subsystem load and configuration. Upon successful loading and initialization of the set, the CLCS users access their software, specific to each engineering discipline, via user logon.

The Activity Management task also provides the capability to request a change (load or reload) to a test configuration (in real time) in response from a request from the Master Function. Additionally, the Activity Management task can obtain integrity and version/revision information of software currently stored on a server or hardware platform, and compare this information to pre-defined test configuration parameters.

3.1.3. Set Configuration and Start Up

Startup tasks and procedures are performed in order to prepare a group of CLCS resources to support testing of a Test End Item (TEI) or simulated component. Based on the outcome of the Resource Manager function, the Set hardware is configured with the appropriate Flow Zone, Control Group, Gateway Group, and Network segments. System integrity monitoring is enabled and the scheduled TCID software and tables are distributed, along with the associated operating system. The Test set is loaded with the software and tables for that Test. Operational readiness is verified and system processes are initialized and started. Connectivity to the proper Test End Item is verified and data acquisition and processing is started.

3.1.3.1. Set Configuration

As many safeguards as possible should exist to ensure proper set configuration. Hardware configurations should be checked and validated by software all the way to the end item for that test configuration. Software should be available to monitor the configuration at various levels including the set, subset (Flow Zone), user positions, and individual components. This software should provide configuration information about the TCIDs loaded in each subset, each user application set, overall resource sizing information, all network components, etc. All configuration information should be recorded.

3.1.3.2. Set Load

In order to reconfigure a set from one TCID to another in a reasonable time, multiple TCID storage and automatic load capabilities are needed. Several versions of system software and application packages (TCID's) may be stored at the SDC. During the pre-load process, the Activity Manager will retrieve the operating system software and the applications software for the appropriate test configuration from the SDC and transfer it to local application servers. Upon successful initialization of subsystems with system software, the application set required to support testing may be made automatically available to the user from the local applications server upon successful log-on.

The loading of software to resources within a set is managed by the Activity Management task. When invoked by the Master Function, this task provides the capability to load an entire CLCS set, a specified group of resources within a set, or an individual resource. When an entire set is to be loaded, a test configuration (which consist of an associated collection of software and hardware resources needed to support a test), is pre-defined and stored in a centralized storage location within the SDC. The software to be loaded consists of the operating system, system applications and user engineering applications, as well as supporting tables, etc., that make up the test configuration software package (TCID). The software test package is built by the software development process and placed under configuration control prior to being loaded into an operational environment. When a configuration is needed for testing, operations personnel retrieve the specific configuration and, upon verification of the desired configuration, initiates the autonomous loading of the specified hardware. The Activity Management task associates a group of gateways, processors, servers, etc., with a software test package as pre-defined by the Test Configuration, in addition to a group of user positions (workstations) to support the test configuration. The Activity Management task monitors the progress of the load process, while the SDC loads the hardware and provides data integrity and verification mechanisms (CRC, checksums, etc.) to insure proper subsystem load and configuration. Upon successful loading and initialization of the set, the CLCS users access their software, specific to each engineering discipline, via user logon. The Activity Management task also provides the capability to request a change (load or reload) to a test configuration (in real time) in response from a request from the Master Function. Additionally, the Activity Management task can obtain integrity and version/revision information of software currently stored on a server or hardware platform, and compare this information to pre-defined test configuration parameters.

3.1.4. Operational Readiness Test Task

The Master Function also supports an Operational Readiness Test task, which provides the capability to safely test system resources. These tests provide a "level of confidence" that a resource is properly configured and ready to support in an operational environment. This task exercises the software and hardware interfaces which are utilized when the resource is functioning in an on-line (operational) environment. Specific restrictions are incorporated such that, testing of resources that are directly connected to a Test End Item, do not adversely effect the Test End Item itself.

3.1.5. Network Manager Function

The Network Manager Function is responsible for the management, configuration, and monitoring of all network components for all networks supporting the CLCS architecture. A Network Manager Function is provided within each OCR to manage these tasks. It provides health and performance data to all Master Functions (for the System Integrity and Activity Management tasks) within its OCR, to the Maintenance Monitor Function for load balancing and system tuning, and to the SDC for notification. As an integrated control and monitoring platform, it interfaces with commercial sniffers and analyzers to provide additional insight into system anomalies and performance bottlenecks.

3.1.6. Data Retrieval and Distribution (SDC)

SDC will provide the capability to retrieve data that has been recorded. There are two types of recorded data: packet data and distributed data (see paragraph 3.2.6 for details concerning these data types). Packet data is temporarily available on-line for a TBD period of time after which it will be saved on tape. Distributed data will be available on-line to the extent possible based upon disk storage capacity and then archived to tape in a format which is easily brought back on-line should retrievals on this data become necessary.

SDC will also distribute the same Distributed Data (not the RTPS packets themselves) as a consolidated data stream. This data will be coded so that the subscriber can monitor the data from a specific source. SDC will also distribute all tables and other information required to effectively decode the data stream. This data stream is distributed via two methods: multi-cast and point-to-point, depending on the network requirements.

Existing retrieval applications which are applicable to CLCS will be useable without modification, however, only the data values with millisecond time accuracy will be reported and not the associated health status. Both packet-type retrievals and distributed data type retrievals can be done from a use written application using a binary socket-level interface to SDC. SDC will provide at least two basic retrievals specifically designed for CSCS data. For packet-type data, SDC will provide a retrieval capability to select a time-span, packet type, and then receive a formatted dump. The dump will consist of decoded packet and payload headers and a hex/ascii dump of the packet contents. For distributed data, SDC will provide a retrieval to report all aspects known about the data including time, value, and health for the selected Function Designators. Eventually, all applicable retrieval applications will be modified to handle the CLCS expanded data.

3.1.7. Simulation

The Simulation System is the portion of the CLCS that provides support for the testing and validation of CLCS equipment; checkout and validation of software used in Shuttle ground testing and launch operations; training of CLCS console operators; and launch team training. To support these activities, a number of master math models will be built to provide integrated simulation support. Each of the master math models will include all outstanding math model updates to support the respective TCID. The master math models, both debug and verified, will be groupings of system models to provide the simulation support for related systems and activities. Debug master math models will provide support for application software development and verification and for each of the Countdown, Cryogenic Loading, and Hypergolic Loading training sessions.

Each new TCID build or edit will require a build or rebuild, respectively, of the master math model for simulation support. The master math model will be built by the software development process prior to being loaded into an operational environment. When a configuration is needed for simulation support, operations personnel will retrieve the specific configuration and, upon verification of the desired configuration, initiate the autonomous loading of the specified hardware.

3.1.8. Maintenance

At the Organizational or System Level, major assemblies are repaired by removing and replacing Line Replaceable Units (LRU) with functional spares. Generally, an LRU is a circuit card, power supply, network or computer box, printer or other assembly whose repair is not practical within the on-line environment. Disposition of any defective unit removed from the operational environment occurs at the Intermediate Level. Custom hardware is repaired within the Intermediate Level Maintenance Facility (ILMF) located at KSC while Commercial Off The Shelf (COTS) hardware may be repaired at KSC or Returned-To-Vendor (RTV) as determined by the Maintenance Plan. Depot Level maintenance is generally performed by the various COTS vendors but may be performed by ILMF personnel. After receiving COTS hardware from the vendor, and after repairing Custom hardware, ILMF performs a thorough retest/validation of all LRUs prior to returning to stock.

3.1.8.1. Organizational Level Maintenance

Maintenance at the Organizational Level is generally limited to the periodic servicing of equipment through scheduled downtimes, troubleshooting to isolate failed components, and the removal and replacement of LRUs and subsystem assemblies. Other actions may include in-place modifications, calibrations, line validations, and various reverification processes. These actions are accomplished through SFOC contractor maintenance personnel only. Any required vendor support within the CLCS program occurs in off-line environments only.

The emphasis at the Organizational Level is to return any failed hardware to an operational state as soon as possible to meet the Operational Availability requirement of 99 percent. The concept of Hot Spares is one of the methods used to meet this goal. The Minimum Peripheral Test Set (MPTS) is a hardware set located within the Launch Control Center which provides a complement of critical spares which are always in use, running diagnostics, or otherwise being exercised. These Hot Spares are immediately available for installation in a control room during critical tests. For less critical support times, Organizational Maintenance relies upon off-line and on-line diagnostic tools, network health and statistics from the Network Manager Function, and operational data from the Master Functions. These data sources supply operational health, status and test result information to the Maintenance Monitor Function.

3.1.8.2. Maintenance Monitor Function

The Maintenance Monitor Function is key to the overall success of the on-line environment in meeting the Mean-Time-To-Repair goal of 30 minutes. It is a multi-faceted tool, dedicated for maintenance use, which provides several capabilities. It serves as a central point to run both intrusive and non-intrusive diagnostic testing on all subsystems associated to any Flow Zone within an Operation Control Room (OCR). Intrusive testing is allowed only on subsystems which have been released from operational support. This diagnostic functionality includes the ability to be used in a debug/monitor mode to capture and log system boot up processes for use in diagnosing subsystem failures. It collects time-tagged operational failure and health data from the Network Manager and Master Functions such as system messages, health and status, network statistics, and Operational Readiness Test (ORT) results. It also has access to historical databases such as PTS, PRACA, and IOS as well as other reference data sources such as the Technical Notification System, and the Maintenance Planning Database. These data sources are used in conjunction with the operational and diagnostic failure data, and analysis software, to provide a comprehensive fault isolation tree to guide maintenance personnel through the troubleshooting process. The end result of this function is an ordered list of the most likely failing LRU(s). This expert/predictive software is also used to provide advisory recommendations to the appropriate Master Function concerning components that have not yet failed completely but are operating in reduced or degraded fashion. This ability allows for timely operational reconfigurations of failing hardware. This high level of integration of operational and diagnostic data sources helps manage maintenance actions to provide the highest level of operational availability to the user.

3.1.8.3. Local Diagnostics

For those systems which are unable to communicate with the Maintenance Monitor Function due to primary network related problems or other faults, local diagnostics which reside on each subsystem are used to fault isolate. These diagnostics are a collection of vendor and custom designed tools, utilities and fault isolation processes. Disk-based, time-stamped error logs and vendor supplied Power On Self Test functions, residing on each subsystem, are used to provide additional insight. Further, connectivity is provided from the console port of each subsystem to the Maintenance Monitor Function to capture boot up failure processes and other messages which may be routed to the port. This ensures that valuable failure data, which would otherwise be lost, is preserved for diagnosis/evaluation.

3.1.8.4. Network Manager Function

Any network related problems are first detected in the Network Manager Function and then passed to the Master Function to allow operational reconfigurations or work-arounds, and to the Maintenance Monitor Function for analysis. The Network Manager Function provides the control, configuration and monitoring capabilities for the supporting network infrastructure within a CLCS set. As it relates to the Organizational Level maintenance activities, the Network Manager Function provides performance and configuration data to the Maintenance Monitor Function for use in load balancing, system tuning, and both proactive and reactive fault isolation and repair. It also provides the control mechanisms for permitting or limiting access by all external network interfaces. Dedicated hardware monitoring test/injection points strategically located throughout the CLCS hardware sets support the connection and use of commercial network sniffers and analyzers as appropriate to the specific network medium. The Network Manager Function, working in conjunction with these commercial tools and other subsystem components, provides an integrated platform from which to base all control, reconfiguration and repair actions consistent with the CLCS Operational & Maintenance Philosophies.

This broad-based approach to Operational Level maintenance which utilizes system performance and failure data, on-line and off-line diagnostics, and powerful software tools is used to meet the requirement to isolate 90 percent of all non-intermittent faults to the LRU level. The CLCS maintenance methodology, in the on-line environment, differs significantly from the original CCMS I approach in that operational data sources are tightly integrated with maintenance and repair functions. This feature greatly enhances the overall system availability.

3.1.8.5. Intermediate Level Maintenance

The Intermediate Level Maintenance Facility provides Intermediate and selective Depot Level maintenance for CLCS equipment using systems and procedures as described in each system's Maintenance Plan. The Maintenance Planning Control Center (MPCC) within the ILMF is the primary Hardware Disposition Area for CLCS hardware and manages the Return-To-Vendor processes. While the repair actions required by some large assemblies may be determined in the MPTS, the disposition process is still managed by the MPCC. Hardware assemblies, repaired by and received from vendors, are verified operational and compatible with existing systems prior to assignment as serviceable spares. Independent verification and validation tasks are performed at the ILMF in a simulated on-line system environment to insure system compatibility of multi-revision level COTS, LRUs and Shop Repairable/Replaceable Units (SRUs). All assemblies are verified prior to use. No LRU or system components will be used in the on-line environment unless they have been verified. This reverification process may occur within the MPTS for large subsystems or for those subsystems which are impractical to transport to the ILMF.

Automatic Test Equipment (ATE) is used to support the repair of any Custom and selected COTS hardware, where practical. Special tools and test fixtures are fabricated, as required, according to the Maintenance Plan to support all repair processes. All products which are routed through the various Intermediate Maintenance Level processes are retested/revalidated prior to returning to serviceable spares.

3.1.8.6. Depot Level Maintenance

Generally, Depot Level Maintenance will be performed by the vendor at the vendor facility. However, there may be circumstances in which these repair actions may be better accomplished at the Intermediate level Maintenance Facility at KSC. The Return-To-Vendor or Repair-On-Site decisions and considerations associated with COTS products are based on the most cost effective method to maintain these products and include:

- warranty considerations
- proprietary components and/or documentation
- test methods and procedures
- revision level management processes
- limitations of on-site resources such as tools, test equipment and skills

3.1.8.7. Summary

The key to successful maintenance in the CLCS, in which Operational Availability is maximized and repair costs are minimized, is based on three principles: First, the early introduction of maintenance planners in the CLCS development process ensures a "maintenance-knowledgeable" design team. Second, a thoroughly researched and developed Maintenance Plan which evolves over the life of the program assures that the correct repair/replace decisions are made for all products and subsystems. And third, a relationship in which operational data sources in the on-line environment are tightly integrated with maintenance processes provides significantly improved system utilization. Adherence to these concepts provides an operable, affordable, and maintainable system.

3.2. Test Operations

Test Operations will include all activities and tasks to perform actual processing of the ground support and vehicle elements.

3.2.1. Test Control

The physical layout of the workstations needs to clearly delineate the grouping. All rooms will be setup to allow the room to be divided. It is intended to group consoles supporting a specific TCID in a designated area within the control room allowing maximum flexibility in the use of hardware and personnel and reducing the number of idle consoles. Consoles will be assigned to a particular TCID for command and control capability.

The Test Set is configured, loaded and initialized. Initializing the Test Set causes an active session with its mapped application processor. Any system generated activity that causes display processor display updates will take place. However, no user interaction with the display is possible prior to logon. The only user interaction the display processor will accept is the logon entry.

3.2.2. System Access

The user will log onto the CLCS console position allowing access to all network resources available to that console position. The TC in charge of the test will receive a screen message that a user has logged on. TC authentication of a logon is only required for remote commanding operations. When a shift change occurs the next user will be able to logon over the previous user. No logoffs should be required and no CLCS actions should occur. All attempts to access the system (successful or unsuccessful) and other security violations will be displayed to the Master Console and/or OTC Console is message form and logged for traceability. Security validation and verification procedures exist for access, command issuance, configuration changes and data integrity.

Each console position is configured with assigned RSYSs. Each user assigned a logon id. Commands are checked for permissions (verified appropriate group is issuing commands). The system software will verify the command appropriately fits within the operation of the test and the command parameters are valid. In order to provide sufficient system security for CLCS, a log-on capability should exist but must be minimized. Safety should not be impeded by access controls. Logon needs to be constructed to provide proper user permission levels, but no preset conditions need to be associated with a particular user id. Physical security should not be any less than what is currently available in CCMS I.

In order to be able to support the additional time required to logon to the console both operations and engineering require that CLCS support:

Single logon allowing access to all network resources available in CLCS.

Users will be allowed to logon to multiple positions (up to 10) and multiple users (minimum of 4) will be able to logon to a single command location.

System messages will continue to be processed after a user logs off and will only be terminated upon test set termination. The user logs off the set using the same mechanism that was used for logon. Test application programs on a test set being used by other users within that test set are not disrupted when a user logs off unless he is the last user to logoff. If the user is the last one in that test set to logoff, the user is prompted to request termination of all test applications. If the user requests termination, all test application programs related to that test set will be terminated. This does not apply to system software applications such as EMON and DMON.

3.2.3. Application Start and System Status

After a successful logon, the user will have access to all applications, available to that console position, to perform a specific task. Any combination of windows, test applications, etc., can be opened and/or executing. Program initialization commands may be initiated by the user via keyboard entry at the dedicated command line area or from a graphical selection mechanism using a keyboard or a pointing device. After application programs are brought up for the appropriate subsystem/task, user will verify health of vehicle subsystem and/or GSE.

3.2.4. Command/Monitor Application

Commands may be initiated by the user via keyboard entry at the dedicated command line area or from a graphical selection mechanism using a keyboard or a pointing device. These graphical selection interactions are interpreted and provided to system software in the same format as if the user had originally entered the command in the dedicated command line area. The system software then can perform all the related syntax and semantic checking regardless of the method utilized by the user to enter the command. After application programs are brought up for the appropriate subsystem/task, user will monitor test operations and respond to system messages, change configuration if necessary, report and solve problems, and terminate test.

3.2.5. Consolidation Efforts

Standard vehicle power up of the INTG, ECLSS, EPDC, ISL and DPS systems, and monitor and control of vehicle power, data and cooling can be performed from a single console position. This baby-sitting position is capable of monitoring for other systems and executing an emergency vehicle power down. A console position of this type will be available for each flow zone.

3.2.6. Command/Monitor Application

Commands may be initiated by the user via keyboard entry at the dedicated command line area or from a graphical selection mechanism using a keyboard or a pointing device. These graphical selection interactions are interpreted and provided to system software in the same format as if the user had originally entered the command in the dedicated command line area. The system software then can perform all the related syntax and semantic checking regardless of the method utilized by the user to enter the command.

After application programs are brought up for the appropriate subsystem/task, user will monitor test operations and respond to system messages, change configuration if necessary, report and solve problems, and terminate test.

3.2.7. Data Recording

The SDC provides a centralized recording capability for time-tagged Shuttle processing data, RTPS configuration changes, error messages, system messages, inter-process communications, and other information. The data is recorded in two different ways: packet recording and data recording. Packet recording provides the capability to record packet-like traffic which occurs on the RTCN and the DCN. This data is recorded with a GMT time-stamp. Data recording breaks apart the packets of data into the individual components (e.g. FD data, messages, etc.). The SDC then sorts, indexes and organizes this data for efficient retrieval. This data is recorded using Data Stream time. When packets from the various subsystems within a Test Set are put out on a RTPS network, the packets are time-stamped by the Gateway) that is being acted upon by that subsystem at any given time. Thus, in order to properly correlate events relating to distributed data, these packets must also be time-stamped using Data Stream time. In support of this, however, the actual GMT when the packet containing the data was placed on the network is also available. Another benefit of the GMT embedded with the recorded data is the ability to easily correlate the recorded data with the recorded packets.

3.2.8. Data Access

Real-time data network access will be available to all areas; including all flow zones, firing rooms, OSB, local sites, and processing sites. There should be no restrictions on shipping data to any NASA or Contractor facility. Access to the command network will be restricted to control rooms and dedicated LC39 local sites only. The business system interface that will be provided must be isolated from the command and real-time data networks. This interface may differ from site to site (e.g., LC-39=SODN, CITE=PON).

3.2.9. Operations Redundancy Definition

For the purpose of this plan the word hazardous means that the loss of CLCS during a test would increase the risk of injury to personnel or damage to hardware.

The entire system will provide a minimum fail safe operational capability. New LPS must provide complete redundancy for command paths before it can be used to support hazardous operations. The only exception to this will be the user display system in the console since adjacent positions will have independent power and data feeds. The system engineers also require that a console operator be notified in a means other than through OIS when his console or application has failed or stopped responding (e.g., the redundant position is notified of the primary position failure).

During transition, the first run of any hazardous or vehicle test using CLCS, old LPS will have the capability to take over the test should a problem occur. Switchability between old and CLCS is assumed and essential in the transition to CLCS to minimize schedule risk.

3.2.10. System Integrity Task

The primary job of the Master Function is to monitor the health and state of a Test Configuration (set). This is accomplished by the System Integrity task within the Master Function. The System Integrity task performs the health and status monitoring of all resources (subsystem and network resources) allocated to a particular test configuration, and performs the redundancy management of all active/standby subsystems. It validates subsystem health, and based on system configuration, commands active/standby switchovers for failed redundant components. The System Integrity task sends messages to the workstation(s) associated to the Master Function for notification and displaying of failed subsystems, or significant events that occur within the set. Selective messages are also sent to all subsystems in the affected set, to the SDC for recording, and to the Maintenance Monitor Function for performance/fault analysis. Additionally, the System Integrity task reports unexpected changes to subsystems (GSE bus errors, PCM format changes, loss of PCM data, etc.). Selective informational and error messages that are received from Subsystem Integrity tasks within all set resources, are processed by the System Integrity task and are also routed to the Maintenance Monitor Function for fault analysis. Redundancy also exist for the System Integrity task, which executes within a separate hardware platform. A graphical user interface provides the capability to monitor both overall system health and status, or detailed subsystem health and status. The network or subsystem status display is selected by the operator to the desired level of detail. For example, the operator can select to see a subsystem's peripheral error counts, or the overall subsystem status. The health and status of each subsystem is also recorded in the SDC.

3.2.11. Subsystem Integrity Task

A Subsystem Integrity task is incorporated within all subsystems, and interfaces with the System Integrity task executing within the Master Function. The Subsystem Integrity task provides health and status information to the System Integrity task for health monitoring and redundancy management decisions, and also records the health and status data to the SDC. Additionally, the Subsystem Integrity task routes error and informational messages to all workstations associated with that subsystem and, on a selective basis, to the associated Master Functions within the set. Each message is tagged as to the set and subsystem that the message originated from, and the time and date of the message occurrence. The message also contains an identifier of the software component originating the message.

3.2.12. Checkpoint Task

An additional task that the Master Function supports the Checkpoint task, which tracks and/or restores subsystem state information to a subsystem. The Checkpoint task provides ongoing (continuous) and snapshot (demand) state information request capabilities, along with interfaces to other task that are utilized to restore a subsystem to a specified state configuration.

4. System Processes

4.1. Hardware

Hardware consists of the task's pertaining to problem identification, resolution and upgrade of the Test Set equipment.

4.2. System Software

System Software consists of the tasks' pertaining to upkeep, upgrade and configuration control of system software and system supplied services/tools software. Included in this is COTS product upgrade/installation and integration.

4.2.1. System Build & Load

System build is identified as the definition, build, and load of target CLCS Sets that are independent of test build. A single system build can support multiple test builds and test build revisions. The definition and build portions are independent of the hardware destination CLCS set. A CLCS Set class defines the scaleable architecture makeup of the set but does not limit the quantity of subsystems in a target CLCS Set.

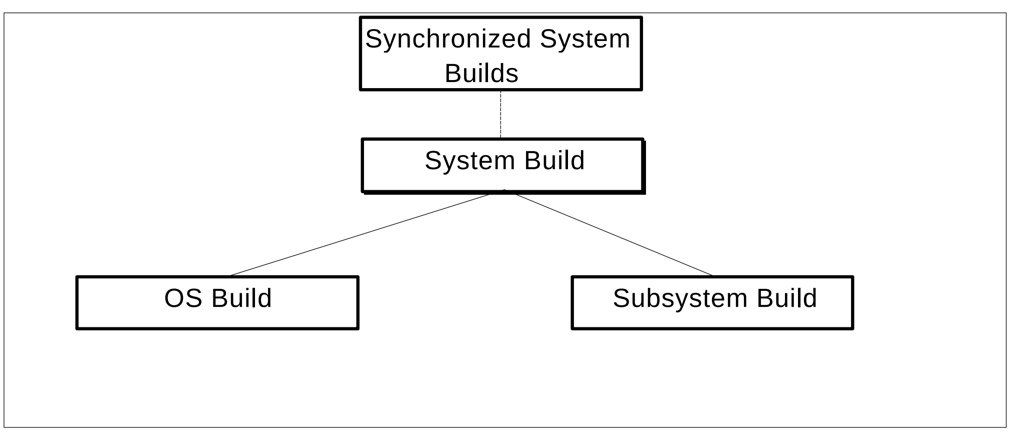


Figure 4-1 System Build Process

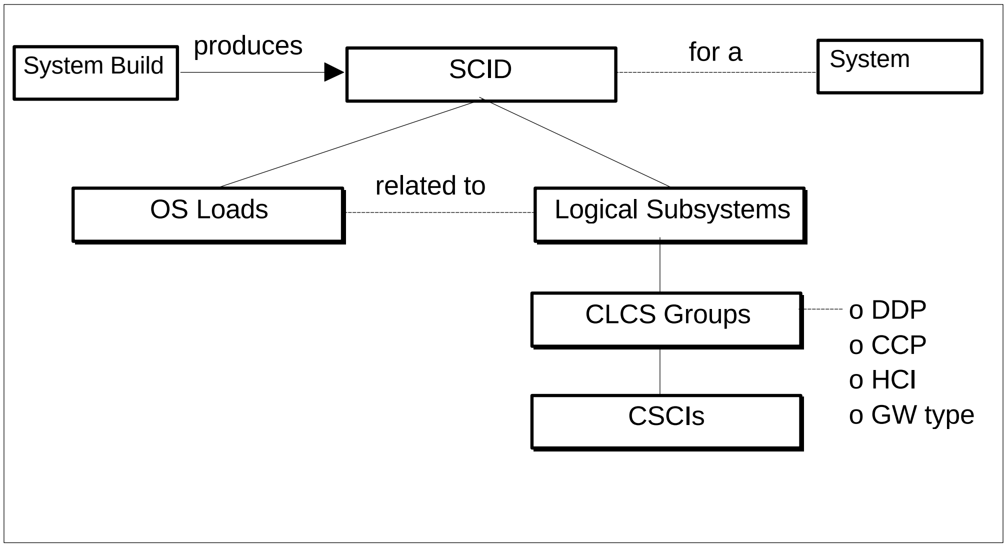


Figure 4-2 Set Build Configuration (SCID) Organization

- Provide OS Build and Load services
- Integrate System Build & Load services with the CLCS Development Environment.
- Support definition/editing of CLCS Set Classes including:
  - CLCS Groups including GW's, DDP, CCP and HCI.
  - Logical Subsystems
    - List of CLCS Groups in the Logical Subsystem
    - Vendor / processor of the Logical Subsystem
- Provide a System Build capability that maintains the Subsystem Builds defines in a CSCS Set Class. A System Build is identified by a SCID (System-build Configuration ID) and a revision.
- Provide a Subsystem Build capability that creates a set of loadable image directories for each subsystem.
- Support parallel maintenance of multiple builds with each system build being a separate revision control thread of underlying components.
- Provide an initial capability to load a System Build onto a CLCS Set.
- Provide System Build support for the following configurations:
  Full Support Configuration
    GW's
    DDP
    CCP
    HCI's
  Limited Support Configuration
    GW's
    DDP/CCP combined
    HCI's
  Application Debug Configuration
    Internal/external model support
    DDP/CCP/HCI combined

Post Redstone Functionality Included For Reference:

- Provide capability to group a maintain a list of Synchronized System Builds representing multiple set configurations.
- Provide a promotion capability

4.2.2. Data Bank-System Build

Data Bank-System Build consists of tasks involving requirements capture, source product identification, engineering assessment, build execution, test/verification execution and release.

4.3. Application Software

Application Software consists of the tasks of requirements capture, engineering assessment, software development (or modification), software validation and requirements close out. This frequently necessitates procedure updates (OMI Rev's).

4.3.1. Simulation Software

The process for developing Simulation Software in the sustaining engineering environment includes all of the traditional aspects of software development, but utilizes them in an iterative and non-serial development cycle. The start of a follow-on phase is not dependent upon the total completion of its predecessor phase.

Requirements Definition: The Requirements Definition phase involves the analysis of the initiating document (e.g., PRCBD, SSCR, RCN, ESR) to define the functional requirements to be supported by the simulation software.

Engineering Assessment: In the Engineering Assessment phase, simulation software design changes required to implement the functional requirements are defined and the cost associated with the change is estimated.

Software Development: In this phase, math modelers develop software to emulate the functional requirements of the hardware modification or replacement. All simulation software development will be performed using the simulation system software development tools and development environment.

Software Debug: Simulation software debug will be performed on all developed and/or modified simulation software prior to being configured to a validation environment. This testing will be performed utilizing a desktop testing environment. Final simulation software debug may be required to occur in the OCR to ensure all system related issues have been addressed. During the software debug phase, the simulation system will support the capability of modifying and loading a new software revision in a minimum timeframe.

Software Validation: Upon successful completion of the debug test effort, the simulation software is validated by the modelers, and users when required, on the simulation system, or if necessary, the Operational Control Room (OCR). This testing ensures the simulation software satisfies the functional requirements.

4.3.2. Command, Control and Monitor (CCM) Application Software

CCM Application Software is categorized into several elements that comprise and application software set:

1. Display Monitors: These monitors will be used to view information about the vehicle/GSE system components and to request, initiate, and control commands and automated sequences.

2. End Item Managers (EIM): EIM will perform all direct FD commanding of vehicle/GSE system components and all automated sequences/tasks. The user will have no visibility into an EIM other than via a display monitor.

3. Data Fusion: Fusion will allow combining values of multiple FD's and/or computations to derive a single value that can be used throughout an application software set. Or across multiple application software sets. The user will specify the algorithm and elements to use for the fusion calculations.

4. Data Path Health: This will provide a mechanism for determining the health of individual FD's. This health will include all data path elements of CLCS (e.g., HIM's, gateways, DDP) as well as the vehicle/GSE specific path elements (e.g., power supplies, signal conditioners, MDM's). The user will specify the algorithm to use for the data path health determination.

The process for developing CCM Application Software in the sustaining engineering environment includes all of the traditional aspects of software development, but utilizes them in an iterative, non-serial development cycle. The start of a follow-on phase is not dependent upon the total completion of its predecessor phase.

1. Requirements Definition **-** This phase involves analysis of the initiating document (e.g. PRCBD, SSCR, RCN, ESR) to define the software functional requirements necessary to answer the request.

2. Engineering Assessment or Design **-** This phase maps the functional requirements into the Application Software Set architecture and existing design. During this phase the design changes necessary for implementing the functional requirements are defined and the cost associated with the change is estimated.

3. Software Development **-** This phase takes the software design and translates it into code using the CLCS defined languages and development tools. Extensive reuse of software components within and across Application Software Sets is employed to help reduce the level of code that must be maintained. All development will be performed off-line from the OCR operations.

    - Display Monitor development will employ a COTS tool for creation/modification of dynamic data visualization displays. This provides a rapid development capability that does not require extensive code development

    - End Item Manger development will employ COTS languages and tools. All Application Software Sets will use the same development tools to ensure compatibility and commonality across the sets.

    - Data Fusion and Data Path Health elements will be developed/modified using System Software provided editors.

    - Debug Testing will be performed on all developed/modified software prior to being configured to a validation environment. This testing will be performed utilizing a desktop test environment that emulates the CLCS system (including DDP, CCP and all of their inherent functions). Upon completion of testing in this environment, final debug testing will be conducted in the Integrated Development Environment (IDE) to ensure all system related issues have been properly addressed. During this testing phase, the system will support the capability of modifying and loading a new software revision in a minimum time-frame.

4. Software Validation - Upon successful completion of the debug test effort, the software is validated by the users in the IDE, which provides a near-operational environment. This testing ensures the software satisfies the functional requirements.

5. Test Procedure Modification - Where necessary, the test procedures (OMIs) that use the newly development software are updated to reflect any change in user interfacing or control mechanisms.

4.3.3. Test Build and Load

A single Test Build can support multiple of Synchronized System Builds.

- Determine whether Test Build should produce memory image tables or an interim form from which memory image tables are created at CLCS Subsystem load/initialization time.
- Integrate Test Build & Load services with the CLCS Development Environment and the SDC.
- Provide GW tables.
- Provide the Online Data Bank Table.
- Provide a Function Designator Directory Table.
- Provide user application files organized by responsible system.
- Provide an initial capability to load and activate a Test Build onto a CLCS Set.

The test build process consists of the functions necessary to allow for the controlling and maintaining of the software required to support launch activities. The test build process includes the following functions: 1)Data Bank Shuttle Function Executive (DBSAFE) Software and 2)Test Configuration Identifier (TCID) Build Software.

The DBSAFE Data Bank provides a repository which describes all of the function designators (FDs) used to support activities at KSC. These FDs are the measurement and stimulus data used to control each individual end item (e.g. valves, switches, and gauges on the Shuttle). This data bank also contains the definition of user generated FDs (e.g. Fusion FDs).

Since the DBSAFE Data Bank supports every possible configuration of measurements and stimulus required at KSC, the TCID Build Software extracts the subset of FD data required to support a specific test or set of tests at KSC (e.g. the FDs relating to the specific tail number of the orbiter support, the ground facilities being used, and the specific FDs for the payload being launched).

This extracted set of FDs are then used to create the necessary files for the CLCS Target Set. The files created are the Online Data Bank which defines for the CCP, DDP, and HCI the function designator data applicable for the test. It also creates the Gateway Tables necessary for the Gateways to read information from the end items and distribute it to the rest of the set.

5. User Environment

The CLCS HCI will replace the basic capability that CCMS provides today, which is to perform test and checkout of the Space Shuttle. CLCS will enhance information gathering, display, and analysis capability through the implementation of many new concepts and new features that will be demonstrated by CLCS in its early deliveries. Other new capabilities will not be provided until later in the development cycle. HCI Information Technology will be integrated into CLCS when it can be purchased off the shelf and provide a design solution for the needs of CLCS.

5.1. Safing

Currently there are four different types of safing in use in the control rooms. They are: LDB safing, GSE hardwire safing, reactive control logic, and program safing through programmable function panels (PFP). Provisions must be made to incorporate the functionality of each type of safing in CLCS. Any proposed new safing systems to support CLCS will be simple, reliable, and support the “generic console” concept.

5.1.1. LDB Safing

The CLCS architecture will significantly reduce the need for LDB safing by eliminating failures common to the current LPS architecture (i.e., console crash or buffer failure). However, an LDB safing equivalent is required for CLCS to protect against command server and command path failures. The preparation and maintenance of the LDB safing software will be part of support operations. The execution of the LDB safing software will be part of test operations.

5.1.2. Hardwire Safing

Currently, loss of LPS can cause loss of GSE or vehicle control capability for critical functions. Since these conditions will still exist in CLCS, hardwire safing panels in the OCR’s must be provided to ensure a backup control system for emergency safing and securing if a significant loss of CLCS occurs. Hardwire safing will provides Appropriate emergency responses and safing actions will be available through a system wholly independent of CLCS and its support systems. The hardwire system will also have a means of continuously verifying its integrity. Sufficient hardwire links will be provided to support both integrated and non-integrated processing from the OCR’s.

Hardwire safing will be implemented in a manner which allows consoles to remain generic. Hardwire safing from CLCS will be connected to the old hardwire safing system running from the LCC to the site. The power for the hardwire safing system must be separate from CLCS and have its own redundancy.

5.1.3. Programmable Function Panel (PFP)

In today’s environment PFP panels are used for sequencing tasks and executing safing sequences. Typically, GLS safing and other critical safing sequences are activated with a PFP keystroke. PFP functionality is necessary but will be used for critical safing operations only. CLCS will provide PFP capability but not necessarily in the same form as today.

5.1.4. Reactive Control Logic

In the current LPS, reactive control logic is used to automatically perform safing/securing operations on predefined out of configuration situations that occur during testing. The functionality of reaction control logic must be preserved under the CLCS architecture.

5.2. Utility Tools

Utility tools include editors, viewers, and specialized command capability both before tests are run as well as during test execution.

5.2.1. End Item Manager

This state tool will allow the engineer to create, edit and view the states of an end item. It will provide a GUI so that a user can easily enter and view information regarding End Item Management. The create and edit function will probably take place in the office well ahead of vehicle testing. The view function will be used real-time in the control room to view End Item Manager configuration.

5.2.2. Data Fusion

Data Fusion involves computations using constants, measurement values, health values or other fusion values. The result of the computation is a value which has a type equal to the data fusion Function Designator (FD) found in the CLCS data bank. Each fusion FD found in the data bank has the same attributes that any other FD of the same type would have with the exception that a Fusion FD does not have a hardware record but does have a fusion algorithm table associated with it. The user may use the CLCS Data Fusion Editor to aid in the input of the fusion algorithm and associated information.

Data Fusion FDs are new FDs created through equations using combinations of old FDs and various algorithms. The Data Fusion Editor and View Tool will allow the engineer to create, edit and view Data Fusion FDs. It will provide a GUI so that a user can easily enter and view information regarding a Data Fusion FD. The Create and Edit Functions of the tool will be used well ahead of vehicle testing. The view function will be used real-time in the control room to view the logic behind Data Fusion FDs.

Data fusion will provide engineers the capability to bring off-line data analysis on-line to the operator real-time as well as reduced software complexity by reducing the need for multiple programs to calculate the same values.. In addition, to further automate subsystem testing we should work to allow certified consolidated data to drive a commanding sequence.

5.2.3. Health

Data Health is based upon a number of parameters, some of which may be external to the system. Data Health is the term applied to the integrity of a Function Designator (FD) value which is being distributed from a CLCS subsystem. It consists of a group of flags which are associated with every FD. Each of the flags is “owned” by different processes within the CLCS system. For example certain flags that deal with the decommutation of the data from its source are “owned” by processes in the CLCS Gateway. Other flags are owned by processes which correlate various data to determine additional “health” information about one or more Function Designators. This information is available to all CLCS processes which utilize FD data.

5.2.4. FD Constraint Check

This tool will allow the engineer to create, edit and view FD constraints. It will provide a GUI so that a user can easily enter and view information regarding FD redline limits and messages. The create and edit function will probably take place in the office well ahead of vehicle testing. The view function will be used real-time in the control room to view the state of the system.

5.2.5. Auto Sequencer

This tool will allow the engineer to create, edit and view sequences of commands. It will provide a GUI so that a user can easily enter and view information regarding command sequences. The create and edit function will probably take place in the office well ahead of vehicle testing. The view function will be used real-time in the control room to view the programmed sequence.

5.2.6. Command Entry

A command entry tool will provide engineers with an ability to enter ad hoc commands to an end item. This tool will employ a Graphical User Interface. The GUI's menus will allow an engineer to choose their system and be presented with a list of their system's FDs. This list can be used to fill out the command entry form. Various safeguards will be implemented in this tool to avoid unsafe command entry.

5.3. Input Devices

5.3.1. Command/Business Network Separation

The Command and Control CPUs will be isolated from the Business and Information CPUs. This isolation enables the user to have access to a broad range of Business and Information Computer Systems with out degrading the Command and Control System's performance or security.

In order to completely separate access to the command and business networks, CLCS will provide keyboards for both networks. Each keyboard may be attached to more than one monitor but can only be connected to one network. Current CCMS has a similar function to switch between CPU (command) and CDS (business) modes. CLCS will provide advantages in this area over today's system. In CLCS when a user switches to the Business and Information CPU, they will still be able to view the Command CPU's screen.

5.3.2. Pointing Devices

A mouse is an acceptable pointing device for CLCS. If there is more than one monitor connected to either the command or business network, the mouse should be able to move between screens to support activities on both.

5.4. Output Devices

Auxiliary ports will be provided at each console with access to power, Business and Information Network, and peripheral devices. There will also be auxiliary ports to accommodate the Launch Back Row Information System. SDC data retrieval capability will be provided from the command monitor at CLCS consoles.

The Common I/O area or Data Review Room within a Control Room should have two combination scanner/fax machines and two black and white laser printers and one color laser printer.

Each console should have one black and white laser printer. There should be one scanner/fax within headset cord reach to allow bringing hardcopy online. "Within headset cord reach" equates to one scanner/fax for two consoles.

5.5. Display Characteristics/Capabilities

The system should only require one set of displays to be built for all command and monitor functions to keep from maintaining more than one display for each function. For example, in the current architecture, the command and control displays, PC-GOAL displays, and ESA displays are all different. In the future, these displays should be one and the same. There will be two distinct type of displays; those associated with the command and control applications and those that are monitor only displays. All displays should be available on any monitor.

CLCS software will provide a Graphical User Interface, (GUI), for controlling every aspect of the user's session. This interface will be built using standard GUI objects such as slider bars, pull down menus, popup menus and dialog boxes. This interface will be easy to learn and work with. The computer operating system should be invisible to the operator. The CLCS GUI will provide a task bar to enable users to quickly identify the applications that are running in their CPU. There will also be an activity indication given for displays that are not visible.

5.5.1. Message Window

The user will have access to a scrolled message window containing some number yet to be determined of program and system messages. This window will be an invaluable tool in providing engineers visibility into the activity of their system.

5.5.2. Overlaying of Windows

Techniques for appropriate window overlaying at an CLCS Command Position will be developed, and corresponding application software implementation standards will be instituted.

5.6. Business Systems

Access to the following Business and Information Network systems should be provided through and integrated information tool. The tool should be World Wide Web compatible.

The Business and Information Network will allow access to the following:
Shuttle Data Center Data Retrievals and Shuttle Data Stream
SL Monitor Displays (Note: same displays as the Command and Control CPU)
PCGOAL Displays in early CLCS
PRACA
Shuttle, GSE and Facility Drawings
OMI's
Goody books, including Pictures and Operational Notes
IWCS (WAD Authoring & Validation Environment, WAVE; ARMS, Shop Floor Control, CASPR)
SCAN/AERI
SECAS/Ground Opns LAN
Office LANS as needed
World Wide Web/Internet
JSC and MFSC Data Systems
Other systems that become necessary (e.g. CCS PLC Controls/Monitor)
Email Support
Xterminal Session to approved NASA Computer Systems (EDAMS, LSDN..etc)
FTP permissions (will be guarded closely)

The Consolidated Data Stream will be made up of HUMS, PMS, Shuttle Data Stream, RPS, Data fusion, and any other data streams utilized in the launch complex area. These data streams will be merged together into a new Shuttle Data Stream. This data will be available everywhere on the network for monitoring purposes.

5.6.1. World Wide Web/Internet Access

The World Wide Web/Internet will be used for gathering information to assist in testing and trouble shooting. Most times this information will be local on a KSC intranet. However, it may be at another NASA center or even a Non-NASA location.

The system will provide the capability to restrict access to and from the control room. NASA security policy regarding World Wide Web/Internet Access will be followed.

5.7. Advisory Systems

Today there are several ‘Advisory’ systems deployed in the CCMS Control Rooms. They have two general classifications; Real Time Data Processing applications and business enhancement applications.

The real-time applications are generally implemented on a high performance dedicated CPU/Platform (e.g. SUN workstation), have a system service that supports independent acquisition and buffering of the PC-GOAL packatized data stream (to the FD Element) and other user unique system services (e.g. CRT Plotting and presentation display) and are USER (or third party) developed and maintained.

In some cases, application specific tools are used for development(AI). (e.g. G2, CLIPS. Etc.). Examples of these types of systems are:

Propulsion Advisory Tool (PSA)

Integrated APU Neural Network AI System (IAPU)

DPS-LCC Expert System (DLES)

Hydrogen Umbilical Monitoring System (HUMS)

The Business enhancement applications are generally a suite of COTS tool with USER unique information/data. (e.g. Databases, sub-system engineering ‘goody books’, Internet tools, etc.)

An example of these applications are the RI-CD-ROM Wirelist Notebook computers and the SSME Avionics Portable computers authorized by the Control Room Enhancement Review Team (CERT) for use in the control rooms.

In order to accommodate these systems in the CLCS design, each advisory system must be reviewed for transition (if applicable) to the CLCS architecture.

One aspect of this capability is the user CM process. Since these systems are ‘advisory’ in nature (augment, do not replace certified systems for decision making) the user has more flexibility and less demanding configuration control requirements. With CLCS, a similar construct is desirable whereby the user can provide his/her own application independent upon the rigor applied to the CLCS Application Software. Additionally this would also require that the CLCS provide a standardized packatized data acquisition and buffering API for the Business Systems platform.

As each existing advisory system is analyzed, a decision must be made to either integrate/port the existing code to the CLCS Application Environment or to ‘Connect’ to the existing system (e.g. in X-Terminal/Moteif fashion). Alternately, a USER Applications Server may be a needed resource in, for example, the SDC. (for user unique business CPU targeted applications).

5.8. OIS

OIS will not be integrated into CLCS. Existing OIS resources including the hardware in the Control Rooms today will be used. There must be space in the console layout to accommodate this OIS equipment. As CLCS matures, OIS could become virtual for a listen only mode.

Standard telephone technology such as call forwarding, beepers, and telephone numbers assigned by system will be used to accommodate the flexible engineering seat assignments afforded by CLCS. As telephone technology improves and provides features such as video teleconferencing, we expect the control rooms to keep pace.

5.9. OTV Displays

OTV will be integrated with CLCS. For viewing, a camera sequencing function will be provided. The user will have the ability to playback OTV, including a slow motion option. Each console command position will be capable of displaying 2 camera views on at least a 20 inch color screen. Camera views are to be displayed in “windows” which will be sizable and moveable. A shared control device will be used to switch views and change window attributes.

5.9.1. View Sequencing

Console operators will be able to view 2 sequences per OTV position. The present ability to sequence through a maximum of 60 views will be maintained. The present ability to adjust the display time between 1 to 60 seconds will be maintained. The capability to define, store and recall up to 12 pre-programmed sequences will be available.

5.9.2. Camera Control

During testing, console operators will have direct control of camera functions (pan, tilt, zoom, focus and lights) through a graphical interface (except consoles in rows OSR and AA). Indirect control of cameras via JYVO (current method) will still be available when requested per OMI (i.e. for major tests, countdown). CLCS applications will be able to status and control camera functions. Camera control permissions will be based on system and test requirements as defined by the users and JYVO. Users will be able to define, store and recall Preset Camera Positions (e.g. press a button and have the camera point in specific direction). Preprogrammed camera movement will be available (e.g. press a button and have the camera automatically scan an area). Preprogrammed camera movement could be tied to Timing and Countdown (e.g. at a specific time in the count have a camera point in a specific direction). Positional feedback (e.g. azimuth and elevation) from cameras will be available but not displayed continuously.

5.9.3. Timing

Timing and Countdown will be overlaid on the OTV recordings with the option to display it.

5.9.4. Special Functions for Recorded Views

Consoles will have capability for real-time hardcopy (to the Shared I/O color printer). Consoles will have capability to start and stop recording of camera views for short sequences (to the Shared I/O area). Video playback with the following options will be available in the Shared I/O Area: Pause, Slow Motion Playback, Fast Forward, Rewind, Frame by Frame Playback, Frame Capture w/ Color, and Hard Copy. JYVO will retain responsibility for archival and long duration recording. Long duration recording requests will be called out by OMI.

6. Operations Scenarios

6.1. Launch/Landing

6.2. Multi-Flow

6.3. Simulations

6.3.1. Integrated Training (S0044, S0056, S0066)

The simulation system will support S0044 (Launch Countdown Simulation), S0056 (Cryogenic Loading Simulation), and S0066 (Hypergolic Loading Simulation) training.

6.3.2. Individual System Training

The simulation system will support individual system training by providing the capability to load up either a single system or integrated model in a standalone training scenario.

6.4. Standalone

6.5. Local Operations

6.6. CCS

PRELIMINARY

04/29/97

CONCEPT OF OPERATIONS
TABLE OF CONTENTS

CONCEPT OF OPERATIONS
TABLE OF CONTENTS